e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 11, November 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.521

6381 907 438 6381 907 438 ijmrset@gmail.com www.ijmrset.com

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI Game Agent with Deep Q-Learning using Reinforcement Learning

T. Rohith, P. Sai Charan, C. Sai Kiran Kumar Reddy, R. Sri Harsha Rao, S. Sai Kiran Reddy, Prof. Sweety Julia

U.G. Student, Department of Computer Engineering (AI&ML), Malla Reddy University, Hyderabad, India

U.G. Student, Department of Computer Engineering (AI&ML), Malla Reddy University, Hyderabad, India

U.G. Student, Department of Computer Engineering (AI&ML), Malla Reddy University, Hyderabad, India

U.G. Student, Department of Computer Engineering (AI&ML), Malla Reddy University, Hyderabad, India

U.G. Student, Department of Computer Engineering (AI&ML), Malla Reddy University, Hyderabad, India

Professor, Department of Computer Engineering (AI&ML), Malla Reddy University, Hyderabad, India

ABSTRACT: Reinforcement Learning (RL) has emerged as a pivotal methodology in developing AI game agents capable of autonomously learning and mastering complex tasks. By leveraging trial- and-error interactions with their environment, these agents optimize their decision-making policies to maximize cumulative rewards. This abstract explores the application of RL in game agents, highlighting key techniques and achievements. One prominent RL algorithm is Q-learning, where agents learn a Q function that estimates the expected utility of taking a given action in a given state. More advanced methods, such as Deep Q-Networks (DQNs), incorporate neural networks to handle high-dimensional state spaces typical in modern video games. Policy gradient methods, like Proximal Policy Optimization (PPO) and Actor-Critic models, further enhance the agent's ability to learn continuous action spaces and adapt to dynamic game environments. The integration of RL in game agents offers several advantages, including improved performance, the ability to learn complex strategies, and adaptability to unforeseen scenarios. However, challenges remain, such as sample inefficiency, the need for extensive computational resources, and difficulties in transferring learned behaviors to new tasks. Future research directions involve enhancing sample efficiency through model- based RL, improving generalization capabilities, and reducing the computational burden of training sophisticated game agents. RL has revolutionized the development of AI game agents, enabling them to achieve superhuman performance in various games. Continued advancements in RL algorithms and computational power promise to further elevate the capabilities and applications of AI in gaming and beyond.

KEYWORDS: Reinforcement Learning, AI game agents, Q-learning, Deep Q-Networks, Policy gradient methods, Proximal Policy Optimization, Actor-Critic models, sample efficiency, model-based RL, generalization, computational resources.

I. INTRODUCTION

Reinforcement Learning (RL) has emerged as a pivotal methodology in developing AI game agents capable of autonomously learning and mastering complex tasks. By leveraging trial- and-error interactions with their environment, these agents optimize their decision-making policies to maximize cumulative rewards. This abstract explores the application of RL in game agents, highlighting key techniques and achievements. One prominent RL algorithm is Q-learning, where agents learn a Q function that estimates the expected utility of taking a given action in a given state. More advanced methods, such as Deep Q-Networks (DQNs), incorporate neural networks to handle high-dimensional state spaces typical in modern video games. Policy gradient methods, like Proximal Policy Optimization (PPO) and Actor-Critic models, further enhance the agent's ability to learn continuous action spaces and adapt to dynamic game environments. The integration of RL in game agents offers several advantages, including improved performance, the ability to learn complex strategies, and adaptability to unforeseen scenarios. However, challenges remain, such as sample inefficiency, the need for extensive computational resources, and difficulties in transferring learned behaviors to new tasks. Future research directions involve enhancing sample efficiency through model- based

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

RL, improving generalization capabilities, and reducing the computational burden of training sophisticated game agents. RL has revolutionized the development of AI game agents, enabling them to achieve superhuman performance in various games. Continued advancements in RL algorithms and computational power promise to further elevate the capabilities and applications of AI in gaming and beyond.

II. RELATED WORK

1. Reinforcement Learning BasicsSutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. This book is a fundamental resource that covers the principles of reinforcement learning, including the theory and algorithms.
2. Deep Q-Learning Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. This paper introduces deep Q-learning, which combines Q-learning with deep neural networks, the technique used in the Snake AI project.
3. PyTorch FrameworkPaszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. This paper discusses the PyTorch library, which is used for building and training neural networks in the Snake AI project.
4. Game AI DevelopmentMillington, I., & Funge, J. (2009). Artificial Intelligence for Games. This book provides insights into the application of AI techniques in game development, relevant to creating the Snake game environment and agent.
5. Previous Implementations and TutorialsPatrick Loeber's Tutorials: The project itself is based on tutorials byPatrick Loeber, which provide practical guidance on implementing reinforcement learning for the Snake game. This includes setting up the environment, agent, andtraining process.

III. METHODOLOGY

1. State Representation:
   Define the game state (e.g., a grid where cells represent the snake, food, and walls).

2. Action Space:
   Define possible actions (e.g., moving up, down, left, right).

3. Reward Function:
   Design rewards for:
   1. Eating food (positive reward).
   2. Colliding with walls or itself (negative reward).
   3. Staying alive for a longer duration (small positive reward).

4. Experience Replay:

Store past experiences (state, action, reward, next state) to sample mini-batches during training.

5. Neural Network Architecture:

Design a neural network to predict Q-values, with layers for input, hidden layers, and an output layer representing Q-values for actions.

6. Training Loop:

Play multiple episodes of the game:

1. For each episode, choose actions based on the current policy.
2. Update the Q-values using the Bellman equation.
3. Store experiences and train the neural network using samples from the replay buffer.

7. Epsilon-Greedy Strategy:

Balance exploration and exploitation during action selection.

© 2024 IJMRSET | Volume 7, Issue 11, November 2024| DOI: 10.15680/IJMRSET.2024.0711108

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

8. Target Network:

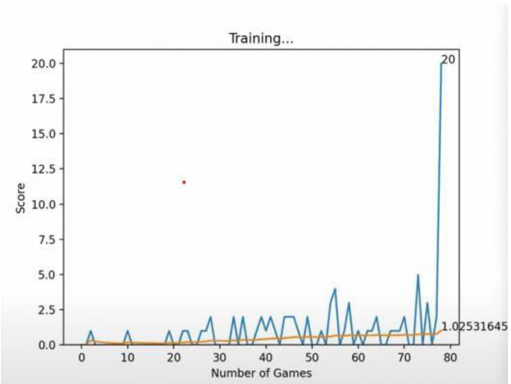
If using DQN, maintain a separate target network that is updated periodically to stabilize training.

9. Performance Evaluation:

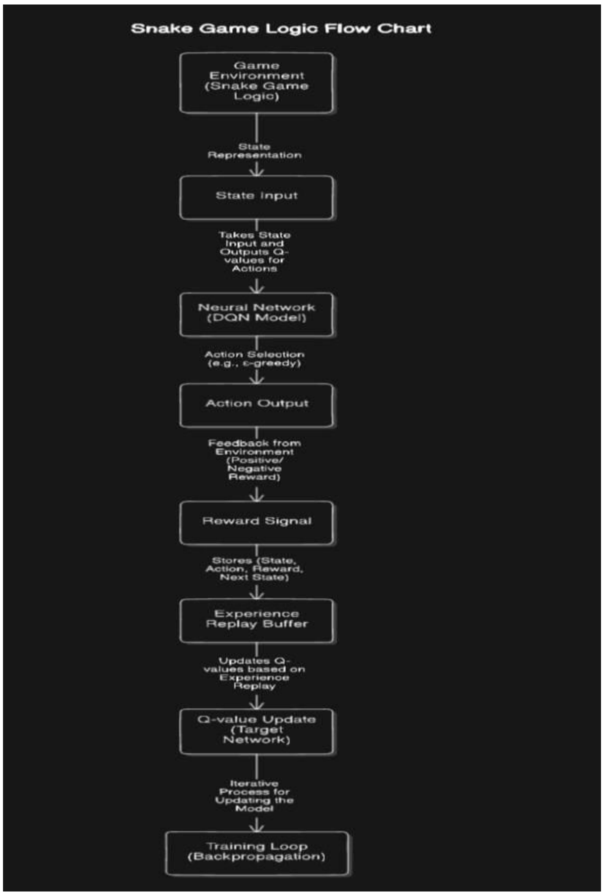
Implement methods to evaluate and visualize the AI's performance (e.g., tracking scores, survival time, and win rates).

10. Hyperparameter Tuning:

Experiment with learning rates, discount factors, and network architectures to optimize the learning process.



ARCHITECTURE

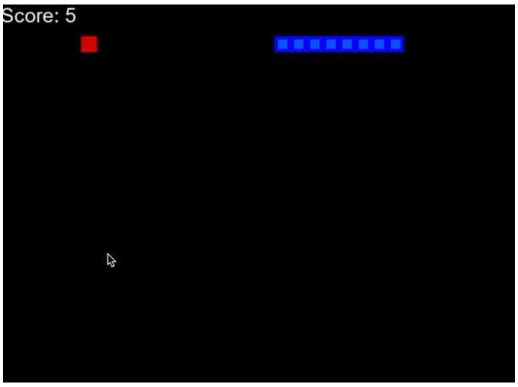


ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. EXPERIMENTAL RESULTS



From the test results we can observe that the snake in the game, learns about its boundaries and get to what step leads to achieve reward and what leads to penalty . It helps balancing exploration and exploitation, and the main objective is to maximize the total reward.

V. CONCLUSION

This project demonstrates a practical application of Reinforcement Learning and Deep Q-Learning for AI agent development. By using PyTorch and Pygame, the AI agent successfullylearns to play the Snake game through trial and error. Key takeaways include:

- Reinforcement Learning:* The agent learns from rewards and penalties, improving its performance over time.
- Deep Q-Learning:* A neural network approximates Q-values, guiding the agent's decision-making. Training Process:* The agent is efficiently trained using the Bellman equation, Adam optimizer, and MSE loss.
- Neural Network Architecture:* A simple feed-forward network processes game states to predict actions.

REFERENCES

[1] Agrawal, R. (1995). Sample mean based index policies with O(logn) regret for the multi-armed bandit problem. Advances in Applied Probability, 27:1054–1078.

[2] Agre, P. E. (1988). The Dynamic Structure of Everyday Life. Ph.D. thesis, Massachusetts Institute of Technology. AI-TR 1085, MIT Artificial Intelligence Laboratory.

[3] Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In Proceedings of the Twelfth International Conference on Machine Learning, pp. 30–37. Morgan Kaufmann, San Francisco.

[4] Greensmith, E., Bartlett, P. L., Baxter,J. (2001). Variance reduction techniques for gradient estimates in reinforcement learning. In Advances in Neural Information Processing Systems: Proceedings of the 2000 Conference, pp. 1507–1514.

[5] Hampson, S. E. (1989). Connectionist Problem Solving: Computational Aspects of Biological Learning. Birkhauser, Boston.

[6] GitHub:https://github.com/patri ckloeber/snake-ai-pytorch

INNO SPACE

SJIF Scientific Journal Impact Factor

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com